Introduction to Machine Learning for Educational Datamining WS2007

Source: Erica Melis

Student talks requirement

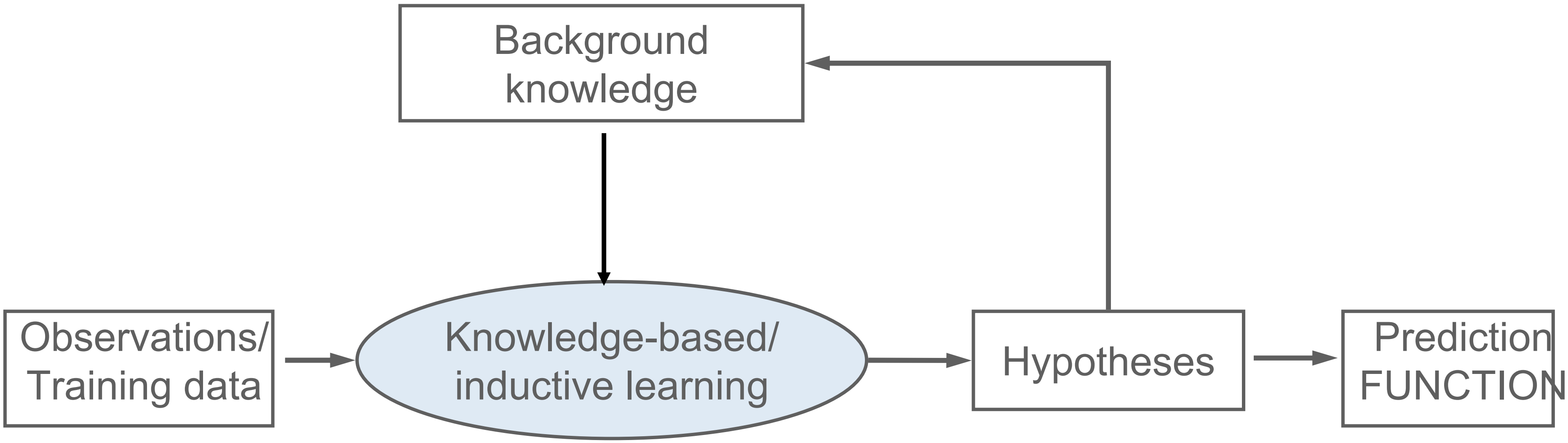
- Present the machine learning technique ‚your' article uses in a short and understandable way (5 - 10 minutes). If a previous talk presented it in a satisfacting way, then go into more detail about ‚your' concrete application.
- Descriptions and software available on the web

Machine Learning Techniques

- **Decision Tree learning**
- Artificial Neuronal Networks
- Instance-based learning
- **Bayes Network reasoning and learning**
- Naive Bayes
- Genetic Algorithms
- Support Vector Machines
- (Hidden) Markov Models
- Reinforcement learning
- Explanation-based learning
- Inductive logic programming
- Boosting

How does an Agent learn? How to choose technique

- Learning goal (output format)
- Input format of training examples
- Amount of input data and background knowledge



Example: Credit Risk Analysis

Customer103: (time=t0)	*Customer103:* (time=t1)	...	*Customer103:* (time=tn)
Years of credit: 9	Years of credit: 9		Years of credit: 9
Loan balance: $2,400	Loan balance: $3,250		Loan balance: $4,500
Income: $52k	Income: ?		Income: ?
Own House: Yes	Own House: Yes		Own House: Yes
Other delinquent accts: 2	Other delinquent accts: 2		Other delinquent accts: 3
Max billing cycles late: 3	Max billing cycles late: 4		Max billing cycles late: 6
Profitable customer?: ?	Profitable customer?: ?		**Profitable customer?: No**
...	...		...

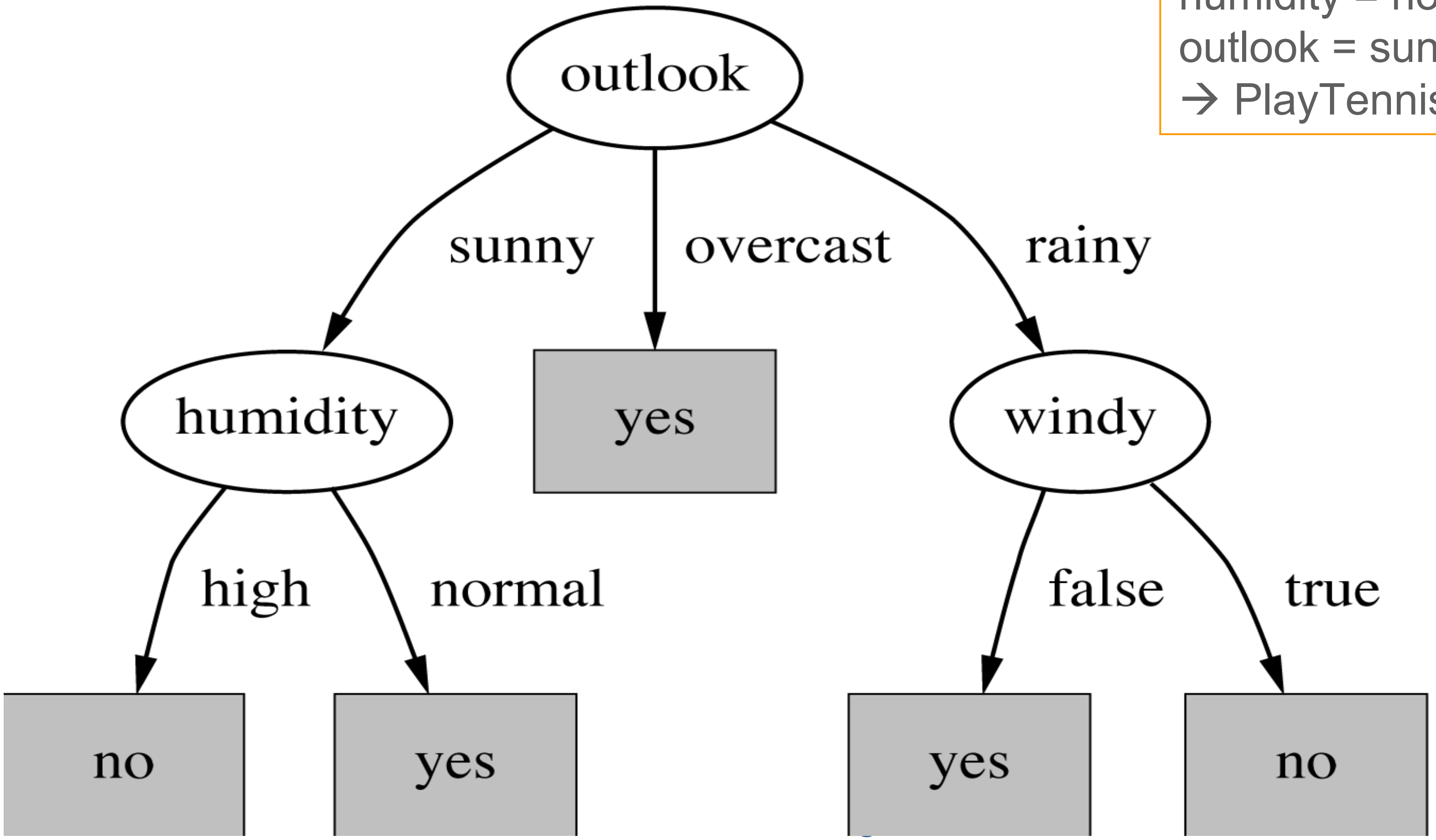
```
If   Other-Delinquent-Accounts > 2, and
     Number-Delinquent-Billing-Cycles > 1
Then Profitable-Customer? = No
     [Deny Credit Card application]

If   Other-Delinquent-Accounts = 0, and
     (Income > $30k)  OR  (Years-of-Credit > 3)
Then Profitable-Customer? = Yes
     [Accept Credit Card application]
```


Machine Learning, T. Mitchell, McGraw Hill, 1997

Decision Tree Learning

Goal predicate: PlayTennis
Hypotheses space:
Preference bias:

temperature = hot &
windy = true &
humidity = normal &
outlook = sunny
→ PlayTennis = ?



Decision Trees: definition

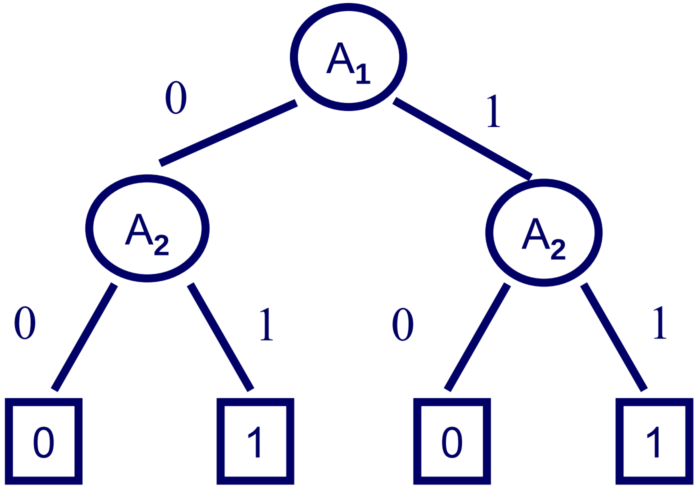
decision tree over the attributes A_1, A_2,.., A_n and G is a tree in which

- each non-leaf node is labelled with one of the attributes A_1, A_2, ..., and A_n
- each leaf node is labelled with one of the possible values for the goal attribute G
- a non-leaf node with the label A_i has as many outgoing arcs as there are possible values for the attribute A_i; each arc is labelled with one of the possible values for A_i

Expressiveness of Decision Trees

Any Boolean function can be written as a decision tree.

A_1	A_2	G
0	0	0
0	1	1
1	0	0
1	1	1



Constructing decision trees

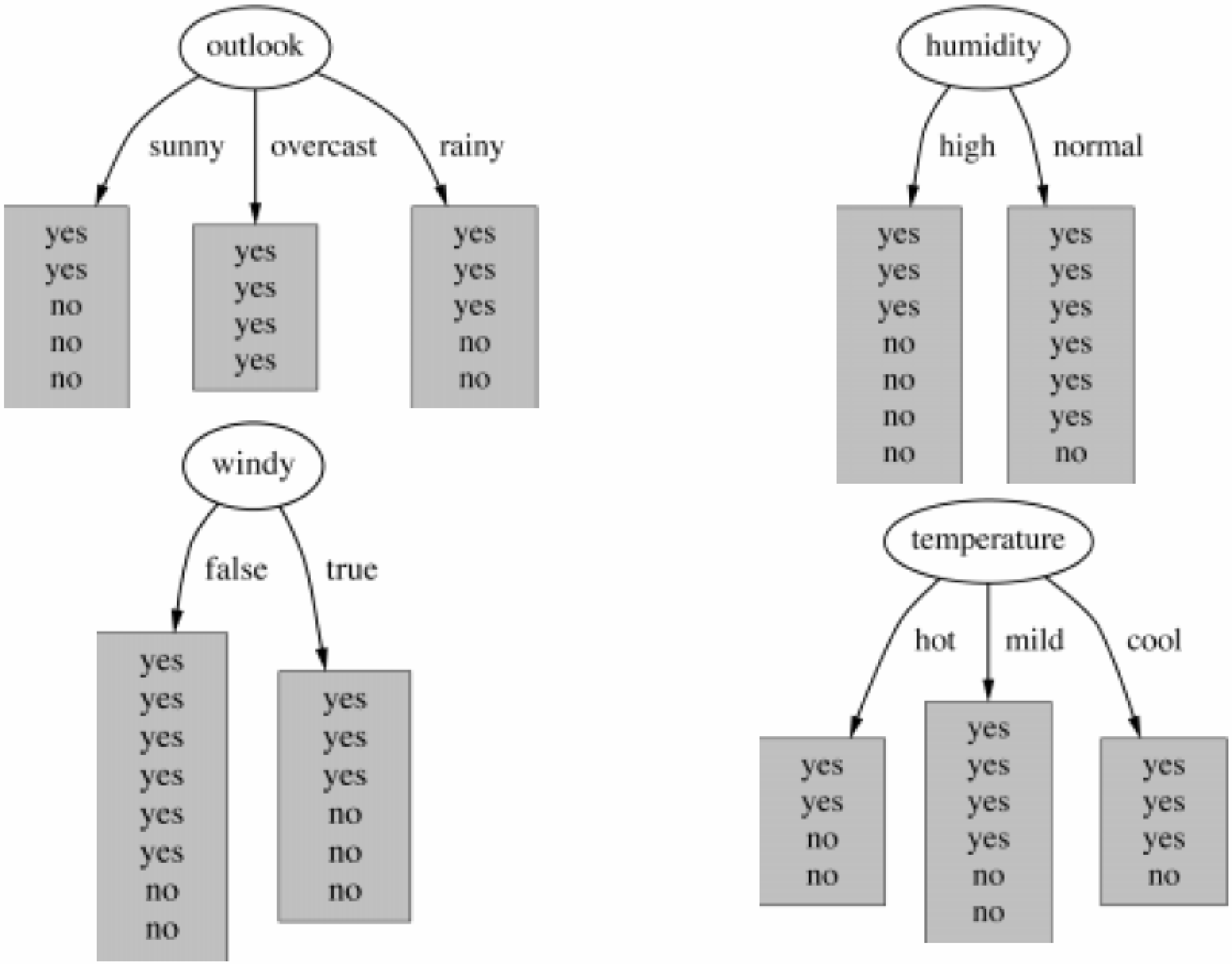
- Normal procedure: top down in recursive *divide-and-conquer* fashion
  - First: attribute is selected for root node and branch is created for each possible attribute value
  - Then: the instances are split into subsets (one for each branch extending from the node)
  - Finally: procedure is repeated recursively for each branch, using only instances that reach the branch
- Process stops if all instances have the same class

Training Examples

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

T. Mitchell, 1997

Which attribute to select?



Computing information

- Information is measured in *bits*
  - Given a probability distribution, the info required to predict an event is the distribution's *entropy*
  - Entropy gives the information required in bits (this can involve fractions of bits!)
- Formula for computing the entropy:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

Example: attribute "Outlook"

- "Outlook" = "Sunny":

$$\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5\log(2/5) - 3/5\log(3/5) = 0.971 \text{ bits}$$

- "Outlook" = "Overcast":

$$\text{info}([4,0]) = \text{entropy}(1,0) = -1\log(1) - 0\log(0) = 0 \text{ bits}$$

Note: this is normally not defined.

- "Outlook" = "Rainy":

$$\text{info}([3,2]) = \text{entropy}(3/5,2/5) = -3/5\log(3/5) - 2/5\log(2/5) = 0.971 \text{ bits}$$

- Expected information for attribute:

$$\text{info}([3,2],[4,0],[3,2]) = (5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971$$

$$= 0.693 \text{ bits}$$

Computing the information gain

- Information gain: information before splitting – information after splitting

$$\text{gain}(\text{"Outlook"}) = \text{info}([9,5]) - \text{info}([2,3],[4,0,[3,2]) = 0.940 - 0.693$$

$$= 0.247 \text{ bits}$$

- Information gain for attributes from weather data:

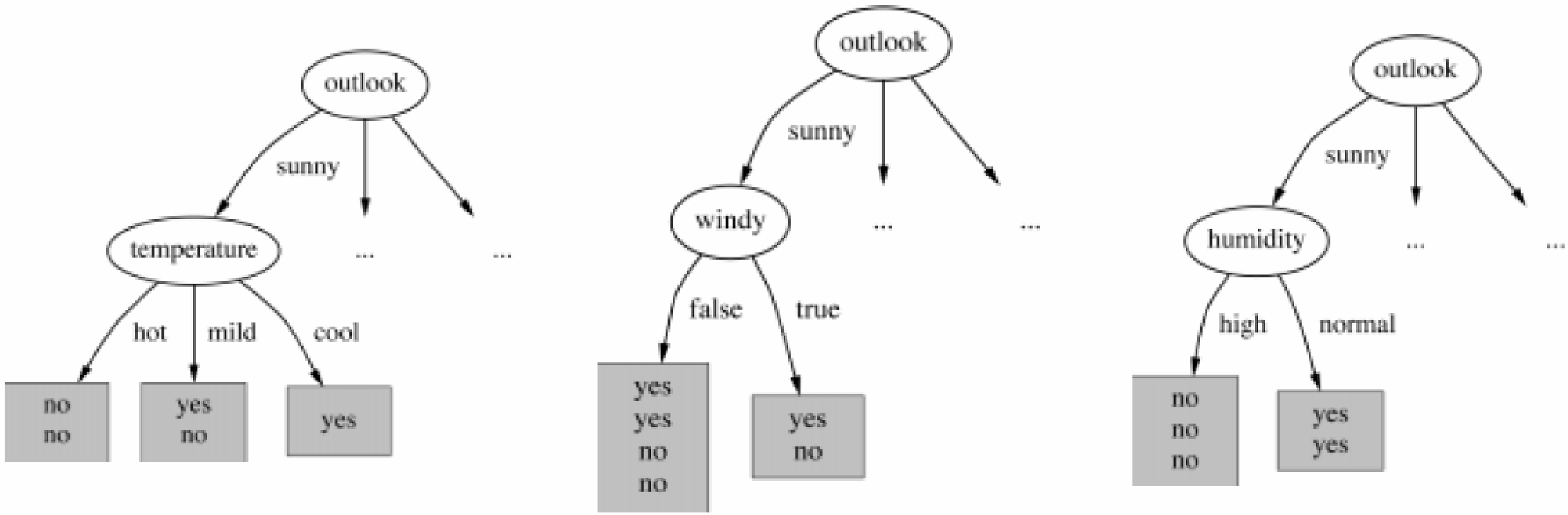
$$\text{gain}(\text{"Outlook"}) = 0.247 \text{ bits}$$

$$\text{gain}(\text{"Temperature"}) = 0.029 \text{ bits}$$

$$\text{gain}(\text{"Humidity"}) = 0.152 \text{ bits}$$

$$\text{gain}(\text{"Windy"}) = 0.048 \text{ bits}$$

Continuing to split

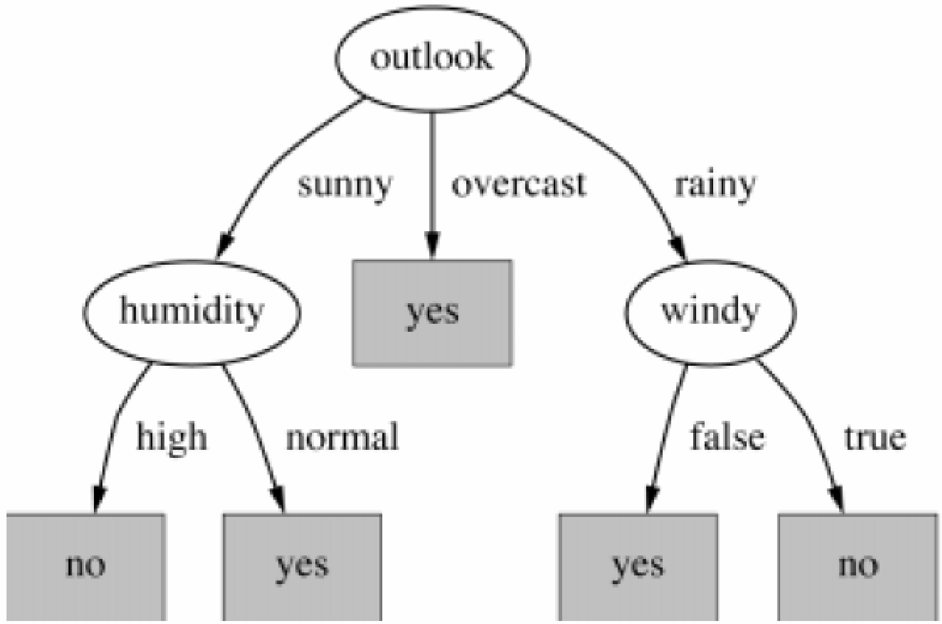


$$\text{gain}(\text{"Temperature"}) = 0.571\ \text{bits}$$

$$\text{gain}(\text{"Humidity"}) = 0.971\ \text{bits}$$

$$\text{gain}(\text{"Windy"}) = 0.020\ \text{bits}$$

The final decision tree



- Note: not all leaves need to be pure; sometimes identical instances have different classes

  $\Rightarrow$ Splitting stops when data can't be split any further

Assessing Decision Trees

a learning algorithm has done a good job, if its final hypothesis predicts the value of the goal attribute of unseen examples correctly

General strategy: cross-validation

1. **collect a large set of examples**
2. **divide it into two disjoint sets: the training set and the test set**
3. **apply the learning algorithm to the training set, generating a hypothesis *H***
4. **measure the quality of *H* applied to the test set**
5. **repeat steps 1 to 4 for different sizes of training sets and different randomly selected training sets of each size**

When is decision tree learning appropriate?

- Instances represented by attribute-value pairs
- Target function has discret values
- Disjunctive descriptions may be required
- Many input data
- Training data may contain missing or noisy data

Bayes Nets Reasoning about Uncertainty

Some slides: courtesy of **Andrew W. Moore**

- **Want to know probability values of a variable *v* given evidences and/or causes**
- **E.g., probability of *motivation = high* given the probabilities of**

  ***mastery = bad* (evidence) and**

  ***self-efficacy =low* (cause)**

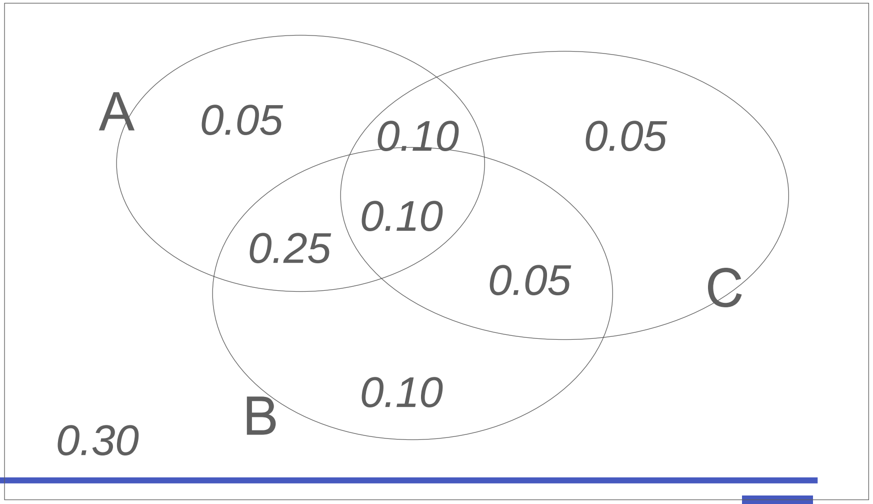
Joint Probability Distribution

Example: Boolean variables A, B, C

Recipe for making a joint distribution of *m* variables:

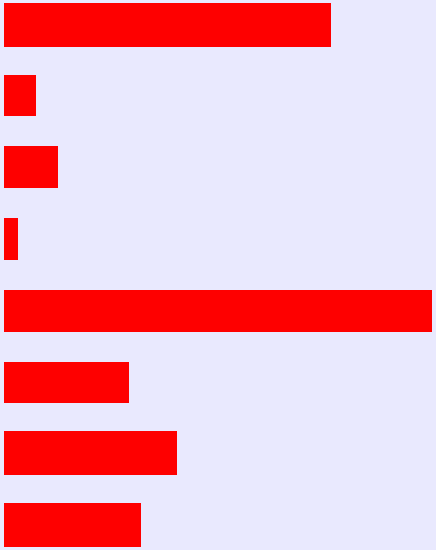
1. Make a truth table listing all combinations of values of your variables (if there are *m* Boolean variables then the table will have 2^m rows).
2. For each combination of values, say how probable it is.
3. If you subscribe to the axioms of probability, those numbers must sum to 1.

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933



Once you have the JD you can ask for the probability of any logical expression involving your attribute

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

P(Poor Male) = 0.4654

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

P(Poor) = 0.7604

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Inference with the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

Inference with the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

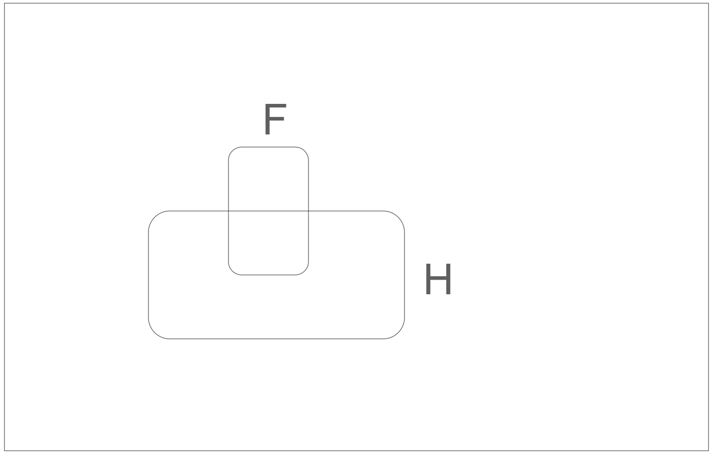
P(Male | Poor) = 0.4654 / 0.7604 = 0.612

Goals

- **Efficiently represent joint Probability Distribution by Bayesian Network**
- **Learn the (structure and tables of) Bayesian Network**
- **Compute probability value of variable in a Joint Distribution by using Bayesian Network**

Conditional Probability

- **P(A|B) = Fraction of worlds in which B is true that also have A true**



H = "Have a headache"
F = "Coming down with Flu"

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

"Headaches are rare and flu is rarer, but if you're coming down with 'flu there's a 50-50 chance you'll have a headache."

Definition of Conditional Probability

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

Chain Rule

$$P(A \wedge B) = P(A|B)\, P(B)$$

Bayes Rule

$$P(B|A) = \frac{P(A \wedge B)}{P(A)} = \frac{P(A|B)\, P(B)}{P(A)}$$

Conditional independence

Suppose we have these three events:

M : Lecture taught by Manuela

L : Lecturer arrives late

R : Lecture concerns robots

Once you know who the lecturer is, then whether they arrive late doesn't affect whether the lecture concerns robots.

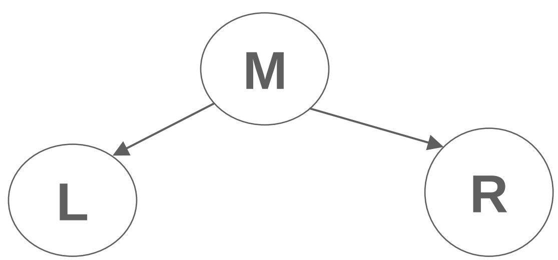
$$P(R \mid M,L) = P(R \mid M) \text{ and}$$

$$P(R \mid \sim M,L) = P(R \mid \sim M)$$

We express this in the following way:

"R and L are conditionally independent given M"

..which is also notated by the following diagram.



Conditional Independence formalized

R and L are conditionally independent given M

if

for all x,y,z in {T,F}:

$$P(R=x \mid M=y \wedge L=z) = P(R=x \mid M=y)$$

More generally:

Let S1 and S2 and S3 be sets of variables.

Set-of-variables S1 and set-of-variables S2 are conditionally independent given

Bayes Net Formalized

A Bayes net is an augmented directed acyclic graph, represented by the pair V , E where:

- V is a set of vertices.
- E is a set of directed edges joining vertices. No loops of any length are allowed.

Each vertex in V contains the following information:

- The name of a random variable
- A probability distribution table indicating how the probability of this variable's values depends on all possible combinations of parental values.

Building Bayes Nets

T: The lecture started by 10:35

L: The lecturer arrives late

R: The lecture concerns robots

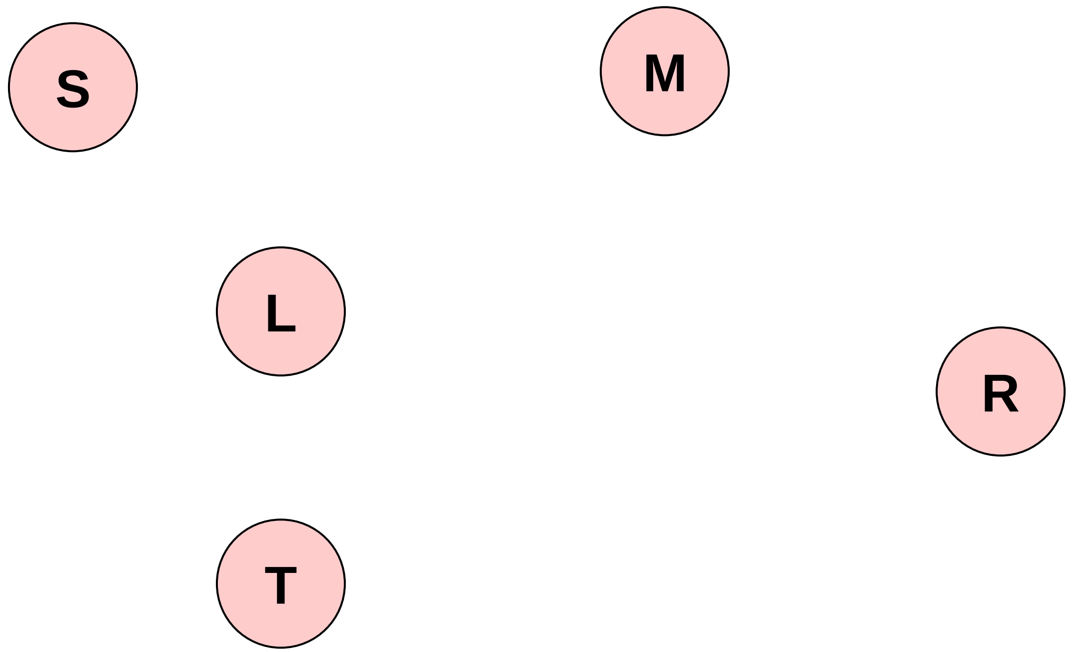
M: The lecturer is Manuela

S: It is sunny

- T is conditionally independent of R,M,S given L
- L is conditionally independent of R given M & S
- R is conditionally independent of L,S, given M
- M and S are independent

Building a Bayes net

T: The lecture started by 10:35
L: The lecturer arrives late
R: The lecture concerns robots
M: The lecturer is Manuela
S: It is sunny

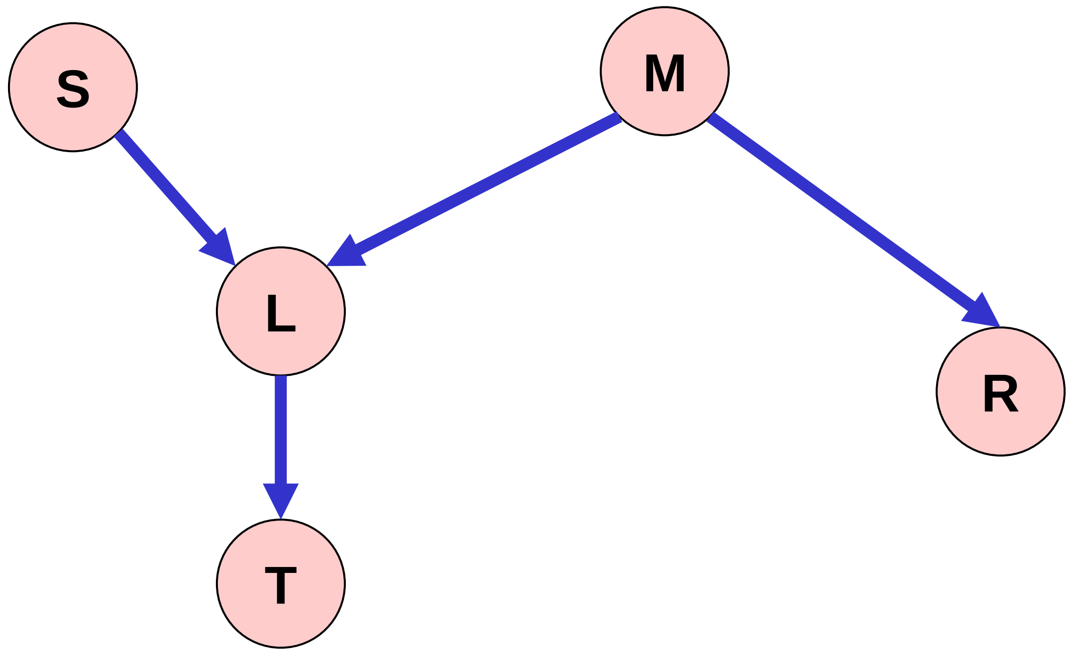


Step One: add variables.

- choose the variables you'd like to be included in the net.

Building a Bayes net

T: The lecture started by 10:35
L: The lecturer arrives late
R: The lecture concerns robots
M: The lecturer is Manuela
S: It is sunny

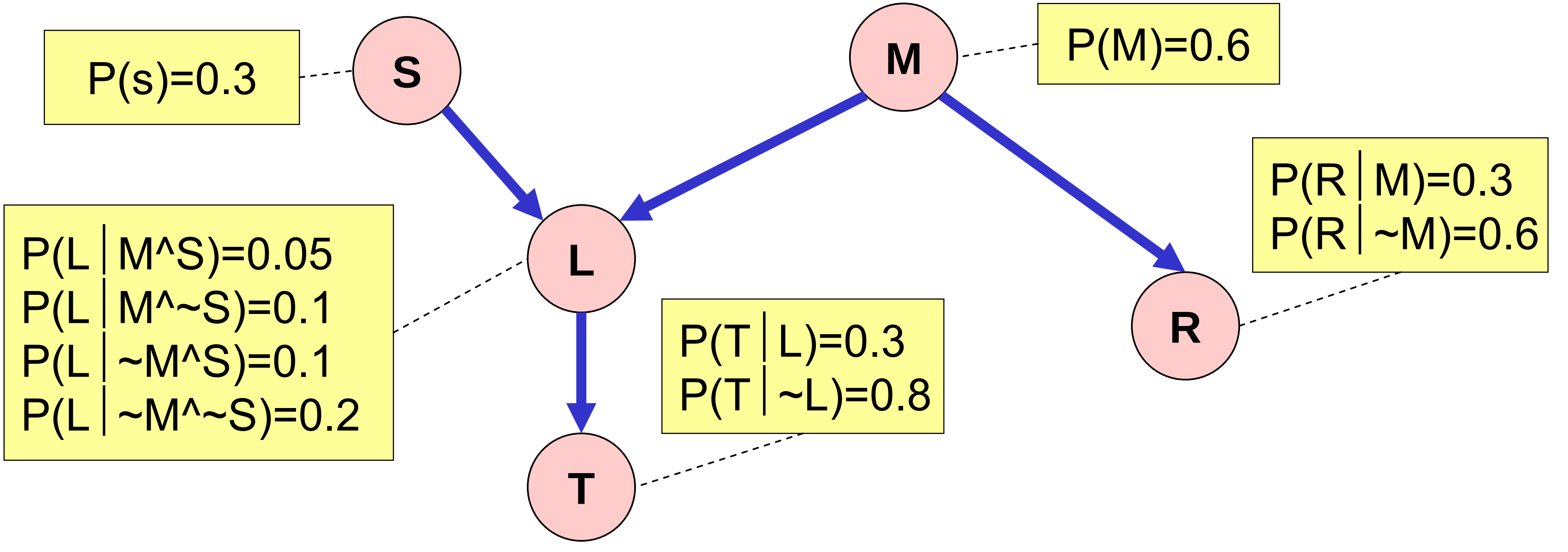


Step Two: add links.

- The link structure must be acyclic.
- If node X is given parents $Q_1,Q_2,..Q_n$ you are promising that any variable that's a non-descendent of X is conditionally independent of X given $\{Q_1,Q_2,..Q_n\}$

Building a Bayes net

T: The lecture started by 10:35
L: The lecturer arrives late
R: The lecture concerns robots
M: The lecturer is Manuela
S: It is sunny

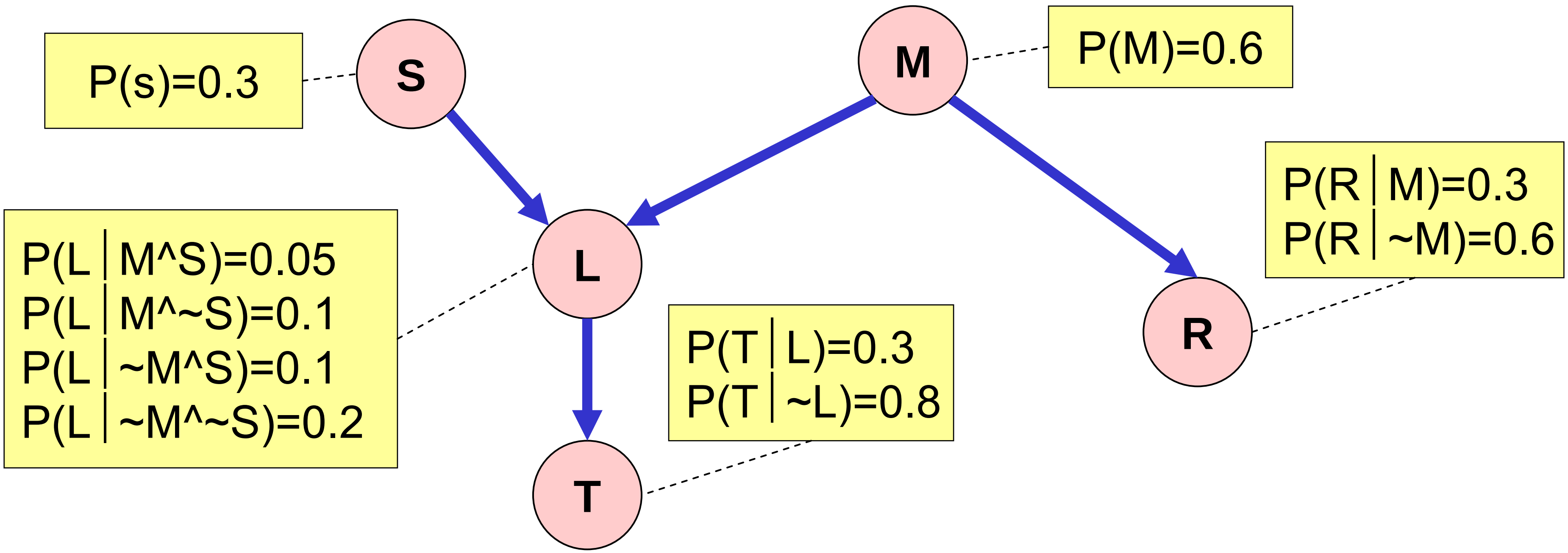


Step Three: add a probability table for each node.

- The table for node X must list P(X|Parent Values) for each possible combination of parent values

Building a Bayes net

T: The lecture started by 10:35
L: The lecturer arrives late
R: The lecture concerns robots
M: The lecturer is Manuela
S: It is sunny

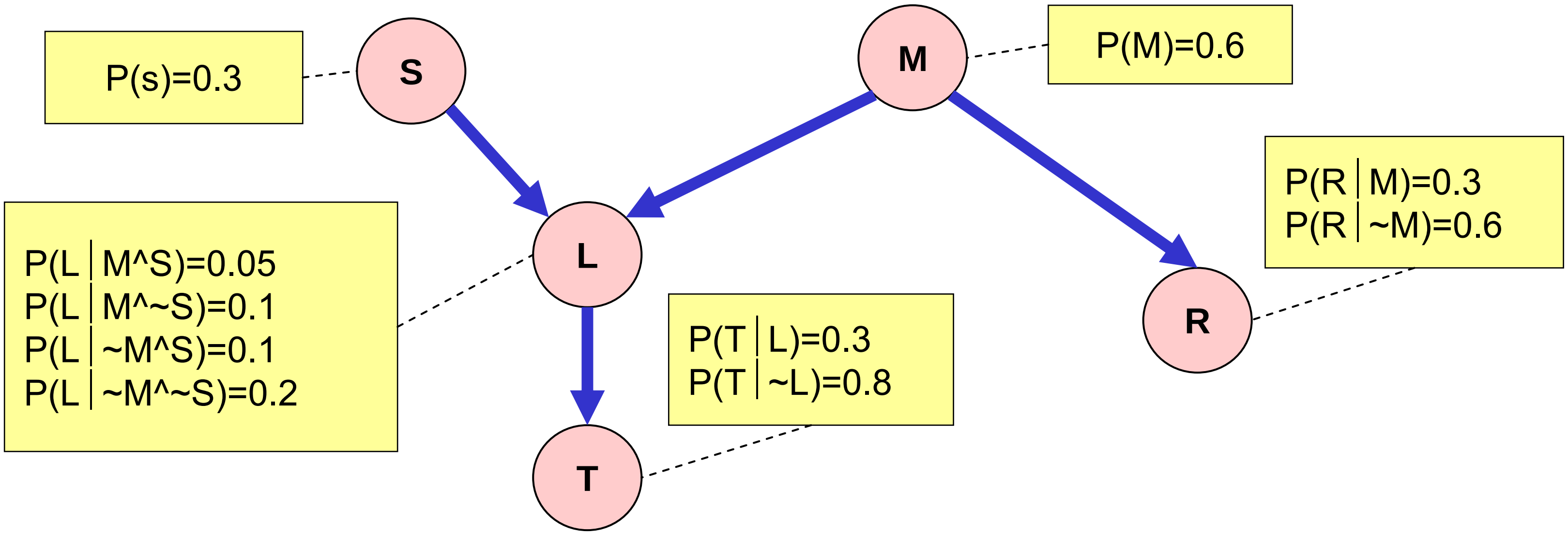


- Each node is conditionally independent of all non-descendants in the tree, given its parents.

Bayes Net Construction

1. Choose a set of relevant variables.
2. Choose an ordering for them
3. Assume they're called X_1 .. X_m (where X_1 is the first in the ordering, X_1 is the second, etc)
4. For $i = 1$ *to* m:
   1. Add the X_i node to the network
   2. Set $Parents(X_i)$ to be a minimal subset of $\{X_1 \ldots X_{i-1}\}$ such that we have conditional independence of X_i and all other members of $\{X_1 \ldots X_{i-1}\}$ given $Parents(X_i)$
   3. Define the probability table of $P(X_i = k \mid \text{Assignments of } Parents(X_i))$.

General Computing with Bayes Net



$$P(T \wedge \sim R \wedge L \wedge \sim M \wedge S) =$$
$$P(T \mid \sim R \wedge L \wedge \sim M \wedge S) * P(\sim R \wedge L \wedge \sim M \wedge S) =$$
$$P(T \mid L) * P(\sim R \wedge L \wedge \sim M \wedge S) =$$
$$P(T \mid L) * P(\sim R \mid L \wedge \sim M \wedge S) * P(L \wedge \sim M \wedge S) =$$
$$P(T \mid L) * P(\sim R \mid \sim M) * P(L \wedge \sim M \wedge S) =$$
$$P(T \mid L) * P(\sim R \mid \sim M) * P(L \mid \sim M \wedge S) * P(\sim M \wedge S) =$$
$$P(T \mid L) * P(\sim R \mid \sim M) * P(L \mid \sim M \wedge S) * P(\sim M \mid S) * P(S) =$$
$$P(T \mid L) * P(\sim R \mid \sim M) * P(L \mid \sim M \wedge S) * P(\sim M) * P(S).$$

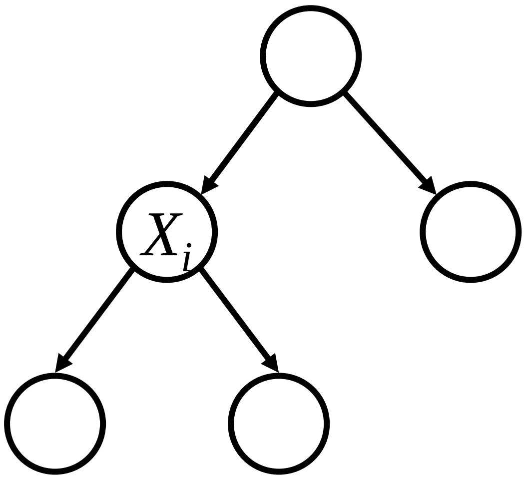
Decomposing the probabilities

$$P(X_i \mid E) = P(X_i \mid E_i^-, E_i^+)$$

$$= \frac{P(E_i^- \mid X, E_i^+) P(X \mid E_i^+)}{P(E_i^- \mid E_i^+)}$$

$$= \frac{P(E_i^- \mid X) P(X \mid E_i^+)}{P(E_i^- \mid E_i^+)}$$

$$= \alpha \pi(X_i) \lambda(X_i)$$

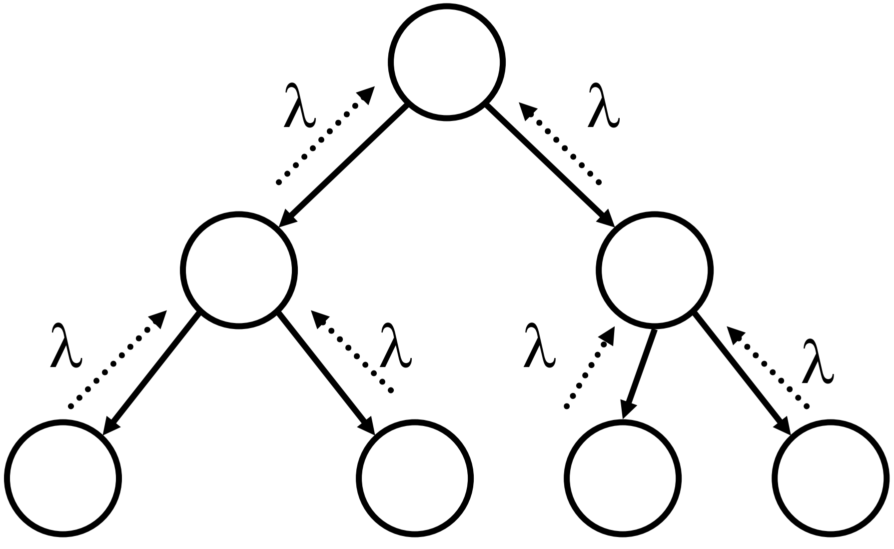


Where:

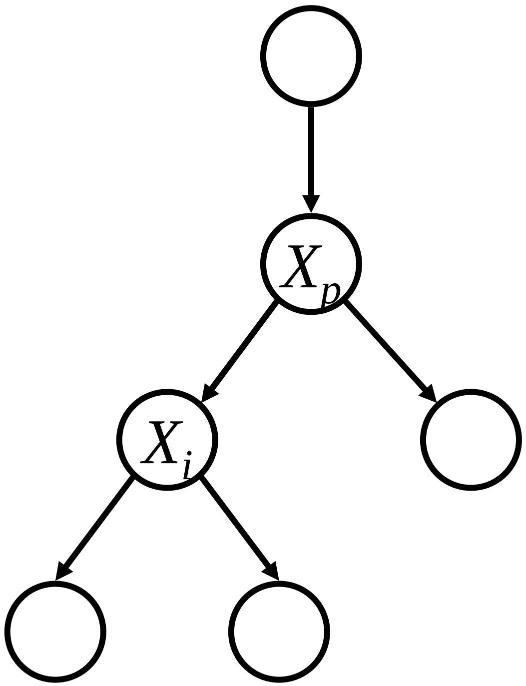
- α is a constant independent of X_i
- $\pi(X_i) = \mathrm{P}(X_i \mid E_i^+)$
- $\lambda(X_i) = \mathrm{P}(E_i^- \mid X_i)$

Evidencial inference

- recursively compute all the λ(X_i)'s, starting from the root and using the leaves as the base case.
- If we want, we can think of each node in the network as an autonomous processor that passes a "λ message" to its parent.



$\pi(X_i) = \mathrm{P}(X_i \,|E_i^+)$ Causal inference



$$\pi(X_i) = P(X_i \mid E_i^+) = \sum_j P(X_i, X_p = j \mid E_i^+)$$

$$= \sum_j P(X_i \mid X_p = j, E_i^+) P(X_p = j \mid E_i^+)$$

$$= \sum_j P(X_i \mid X_p = j) P(X_p = j \mid E_i^+)$$

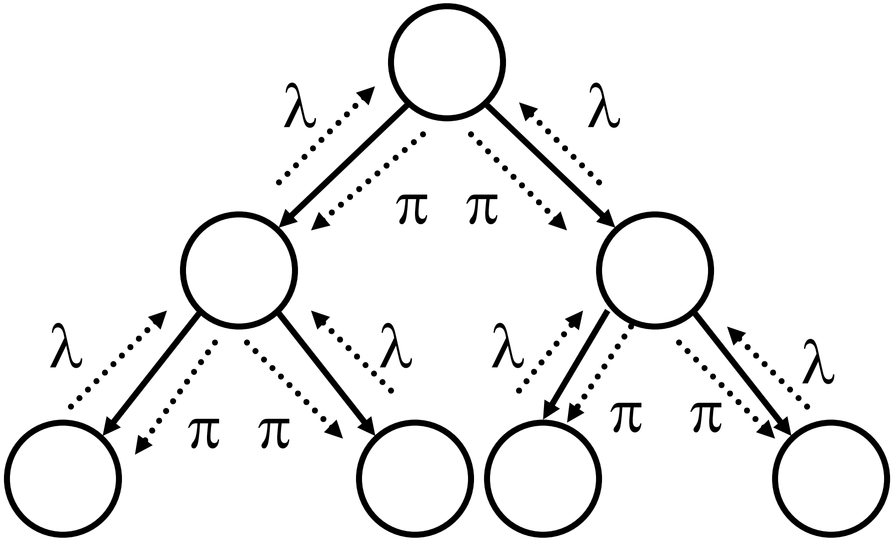
$$= \sum_j P(X_i \mid X_p = j) \frac{P(X_p = j \mid E)}{\lambda_i(X_p = j)}$$

$$= \sum_j P(X_i \mid X_p = j) \pi_i(X_p = j)$$

- For root nodes, X_r , E_r^+ is null set, so $\pi(X_r) = \mathrm{P}(X_r)$.

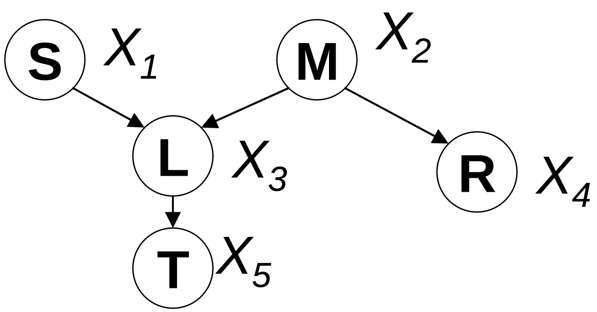
We're done!

- we can recursively compute all the $\lambda(X_i)$'s and $\pi(X_i)$'s, hence all the $P(X_i|E)$'s.
- Can think of nodes as autonomous processors passing λ and π messages to their neighbors

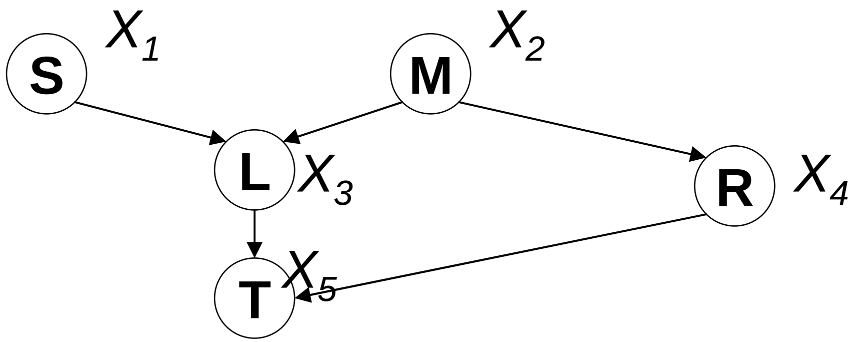


Bayes nets inference algorithms

A poly-tree is a directed acyclic graph in which no two nodes have more than one path between them.



A poly tree



Not a poly tree
(but still a legal Bayes net)

- If net is a poly-tree, there is a linear-time algorithm
- The best general-case algorithms convert a general net to a poly-tree (often at huge expense) and calls the poly-tree algorithm.
- Another popular, practical approach (doesn't assume poly-tree): Stochastic Simulation.

Learning of Bayesian Networks

- **Learn structure**
- **Learn conditional probability tables**

Many applications of Bayes Nets

- A clean, clear, manageable language and methodology for expressing what you're uncertain about
- Already, many practical applications in medicine, EDM, helpdesks:
  - P(this problem | these symptoms)
  - anomalousness of this observation
  - choosing next diagnostic test | these observations

Active Data Collection

Inference

Anomaly Detection